



VAASAN AMMATTIKORKEAKOULU  
VASA YRKESHÖGSKOLA  
UNIVERSITY OF APPLIED SCIENCES

Mathias Renlund

# ORDERBOKFÖRING OCH RESURSÖVERSIKT SOM WEBBTJÄNST

Företagsekonomi och turism

2010

## VASA YRKESHÖGSKOLA

Utbildningsprogrammet för informationsbehandling

### ABSTRAKT

Författare	Mathias Renlund
Lärdomsprovets titel	Orderbokföring och resursöversikt som webbtjänst
År	2010
Språk	svenska
Sidantal	52 + 3 bilagor
Handledare	Kimmo Paulaharju

---

Detta lärdomsprov behandlar utvecklingen av ett webbaserat system för hantering av orderbokningar och resurser åt företaget Jetta Talo Oy.

Lärdomsprovet består av en teoretisk och en praktisk del. Som teoretisk del av rapporten redogörs de webbutvecklingstekniker och koncept som ligger till grund för den praktiska delen där planeringen och utvecklingen beskrivs.

Läsaren får ta del av vilka verktyg som använts under utvecklingen, vilka systemets beståndsdelar bestod av och hur de praktiskt utvecklades.

Programmet är utvecklat i programmeringsspråken JavaScript och PHP, som databas tillämpas MySQL.

Resultatet av detta arbete är ett väl fungerande system som tagits i bruk av beställaren. Avslutningsvis diskuteras resultatet, arbetsgången och förslag till vidare forskning. I diskussionsdelen lyfts också iakttagelser fram som uppdagats under arbetets gång.

VAASAN AMMATTIKORKEAKOULU  
UNIVERSITY OF APPLIED SCIENCES  
Utbildningsprogrammet för informationsbehandling

## ABSTRACT

Author	Mathias Renlund
Title	Order Booking and Resource Overview as a Web Service
Year	2010
Language	Swedish
Pages	52 + 3 Appendices
Name of Supervisor	Kimmo Paulaharju

---

This thesis describes the development of a web-based system for the management of order booking and resources for a company called Jetta Talo Oy.

The thesis consists of a theoretical and a practical section. As the theoretical section of this thesis, the underlying web development concepts and techniques were described. The practical study focused on system planning and the implementation of the case study. The reader is also made familiar with the tools used during the development, and every individual part of the system is described in more detail.

The programming languages used during the developed were JavaScript and PHP. As database manager, MySQL was chosen.

The result of this work is a fully-functional system that has been taken into use by the client. The discussion section of the thesis includes thoughts on the results, the work process and suggestions for further research.

---

Keywords	System Development, PHP, JavaScript
----------	-------------------------------------

# INNEHÅLL

## ABSTRAKT

## ABSTRACT

1	INLEDNING.....	7
1.1	Bakgrund.....	7
1.2	Uppdragsgivare .....	7
1.3	Uppdrag.....	7
1.4	Syfte och mål.....	8
1.5	Avgränsningar .....	9
2	TEORETISK BAKGRUND .....	10
2.1	Introduktion till vTiger CRM.....	10
2.2	Skapande av moduler i vTiger CRM .....	11
2.3	HTML .....	11
2.4	CSS.....	12
2.5	JavaScript.....	14
2.6	AJAX .....	15
2.7	PHP .....	17
2.8	Databas – MySQL.....	19

3	PLANERING .....	21
3.1	Specificering av krav.....	21
3.2	Målgrupp .....	22
3.3	Användargränssnitts design .....	23
3.4	Databas design .....	24
3.5	Skottår och veckoantal .....	25
4	GENOMFÖRANDE .....	26
4.1	Verktyg.....	26
4.1.1	Notepad++.....	26
4.1.2	Paint.NET.....	26
4.1.3	FileZilla FTP.....	27
4.2	Utvecklingsmiljö.....	27
4.3	Grafik .....	27
4.4	Förverkligande av funktioner.....	29
4.4.1	Navigeringsfunktionalitet för ändring av årtal.....	32
4.5	Reserveringspanel .....	33
4.5.1	Borttagningsfunktion .....	34
4.5.2	Bläddringsfunktion.....	35

4.5.3	Spara och Avbryt.....	35
4.5.4	Tillgängliga och otillgängliga fält.....	36
4.6	Översiktspanel.....	37
4.6.1	Datumstämpel för senaste uppdatering .....	37
4.6.2	Excel rapportfunktion.....	38
4.6.3	Stapeldiagram.....	38
4.6.4	Kapacitetsgräns .....	39
4.6.5	Låsfunktion .....	39
4.6.6	Kundkatalog .....	40
4.6.7	Arbetstids kalkylator .....	41
4.6.8	Flyttningsfunktion .....	45
4.7	Säkerhet.....	46
4.7.1	Autentisering.....	46
5	RESULTAT .....	47
6	DISKUSSION.....	48
6.1	Iakttagelser under arbetsprocessen.....	49
6.1.1	Problem med teckenuppsättningar .....	49
6.1.2	Variation i datumhantering mellan olika plattformar.....	49

7	KÄLLFÖRTECKNING.....	51
---	----------------------	----

## BILAGOR

### **FÖRTECKNING ÖVER BILAGOR**

Bilaga 1. Undersökning av datumhantering i PHP

Bilaga 2. Funktion för datumintervall

Bilaga 3. Helhetsbild av det färdiga systemets användargränssnitt.

# **1 INLEDNING**

## **1.1 Bakgrund**

Från och med Maj 2009 har jag parallellt med mina studier arbetat som projektanställd programutvecklare för Vasa yrkeshögskola Ab. Ett företag vid namn Jetta Talo Oy har varit en kund åt skolan, och jag har tillsammans med min projektledare och en arbetskollega infört ett kundhanteringssystem i deras verksamhet, samt byggt upp ett antal skräddarsydda moduler. Kundhanteringssystem kallas också CRM (Custom Relationship Management). Systemet som tillämpats i det här fallet heter vTiger CRM och är baserat på öppen källkod. Mer om detta i kapitel 2 i detta arbete. Jetta Talo har varit nöjd med sitt kundhanteringssystem och har med jämna mellanrum beställt tilläggs funktionalitet kring detta, eftersom inget CRM system kan täcka alla behov för alla verksamheter utan modifikation.

## **1.2 Uppdragsgivare**

Uppdragsgivare för detta arbete är Jetta-Talo Oy. Företaget erbjuder egnahemshus, garage och radhus i elementutförande. Företaget är en av de främsta inom den Finska byggnadsbranschen. Jetta-Talo koncernen består av två bolag; Jetta-Talo Oy och Jetta-Puu Oy. Företagets omsättning var år 2008 14,6 miljoner euro. Företaget sysselsätter idag 120 personer.

Företagets verksamhets ort är kommunen Perho, som är beläget i Mellersta Österbotten. (Jetta-Talo webbsida, 2010)

## **1.3 Uppdrag**

Jetta-Talo Oy har behov av ett centraliserat system för leveransplanering och resurshantering för sin försäljning av elementbyggnader. Problematiken ligger i att företagets försäljare som är utspridda över landet, i dagsläget saknar möjlighet att på



ett effektivt sätt hålla reda på hur mycket resurskapacitet mätt i arbetsdagar som är reserverade, och vilken mängd resurser som finns tillgängliga vid en viss tidpunkt. Systemet skall fungera som en veckokalender där företagsledningen och försäljarna kan få en helhetsbild över hela årets leveranskapacitet och ha möjlighet att reservera sina kunders beställningar. Detta system skall ingå som en modul i det befintliga CRM:et. Språket för användargränssnittet är finska.

#### **1.4 Syfte och mål**

Syftet med detta arbete har varit att utveckla en fullständig webbtjänst som uppfyller beställarens önskemål. Systemet skall ge företagsledningen och försäljningsavdelningen ett centraliserat system för hantering av orderbokningar och de resurser som krävs för att leverera hus som blivit beställda.

Målet var att skapa en webbtjänst som är överskådlig, lättnavigerad och framtidssäker; programmet skall fungera i de vanligaste webbläsarna såväl idag som imorgon.

Rent programmeringsmässigt, eftersträvades en logisk indelning av koderna, där funktioner separeras från övrig kod. En annan viktig aspekt var kommentering av koden, som gjordes på engelska istället för på svenska, med det i åtanke att andra programmerare skall kunna läsa och förstå koden ifall systemet skall vidareutvecklas i framtiden.

I rapporten eftersträvades en balans mellan programmeringsteknisk beskrivning och funktionell beskrivning, för att framhäva kopplingen mellan teori och praktisk tillämpning.

## **1.5 Avgränsningar**

I denna rapport redogörs varken kundhanteringssystemet Vtiger:s uppbyggnad på kodnivå eller databasdesign, även om det krävs förkunskaper i detta för skapandet av skräddarsydda moduler.

Införande och uppföljning av systemet ägnas inte någon uppmärksamhet utan fokus ligger på utvecklingen av systemet och de tekniker som ligger till grund för det.

## 2 TEORETISK BAKGRUND

I detta kapitel tillhandahålls fakta och exempel kring de tekniker och principer som använts vid utvecklingen av denna webbtjänst. Teknikerna presenteras i en naturlig ordning, där grundläggande tekniker kommer först, efterföljt av vidare tillämpningar.

### 2.1 Introduktion till vTiger CRM

Ett CRM är som namnet säger, ett kundhanterings system. Syftet med ett CRM är att hantera företagets kundrelationer, vilket i praktiken betyder att varje kund lagras som en entitet i ett datorsystem, vanligtvis i en databas, och så knyts information till denne. I CRM systemet lagras sådan information om kunderna som är av stor betydelse ur affärsverksamhets synvinkel.

Det finns ett stort antal mjukvaruföretag som erbjuder CRM-system, de främsta leverantörerna inom området är *SAP*, *Salesforce* och *Oracle*. (Gartner, juli 2009) Det som skiljer vTiger från andra CRM är att vTiger är baserat på öppen källkod och kan därmed laddas ned helt utan kostnad. (vTiger webbsida, 2010)

vTiger innehåller en bred samling hanteringsverktyg som kretsar kring kunden. Kunderna uppstår vanligen som en s.k. ”Lead”, som betyder *potentiell kund* eller kan även kallas *prospekt*. En lead kan uppstå utifrån ett kontaktformulär hos den säljande partens hemsida, eller t.ex. namnlistor insamlade under en mässa eller andra typer av PR-evenemang. Lead:s följs upp av försäljare och en viss procent av dessa blir kunder för företaget. Processen mellan lead:s och kund som slutför ett köp, är ett pärlband av handlingar. Alla steg från lead till genomfört köp knyts till kunden i CRM-systemet. Med insamlad information, över alla lyckade och misslyckade köp, kan man sedan göra upp statistik och företaget kan lära sig att optimera sin försäljning. Lagerbokföring, och kundportal för supportmöjligheter ingår också som en del av CRM systemet vTiger.

## 2.2 Skapande av moduler i vTiger CRM

Ifall en viss funktionalitet i CRM systemet efterfrågas, som inte finns från förut, så finns det möjlighet för programutvecklare att skapa egna funktioner. CRM:et i sig, är uppbyggt kring moduler, tanken bakom detta är att systemet skall vara skalbart, moduler som inte behövs skall kunna inaktiveras.

För att skapa egna moduler i CRM:et vTiger finns ett särskilt verktyg för ändamålet. Verktöget heter vtlib och fungerar som ett ramverk för utveckling av skräddarsydda moduler. Vtlib innehåller ett flertal API (Aplication Programming Interface) som säkerställer att skapandet av modulerna går till enligt ett visst mönster och i rätt ordning. (vTLib, 2010)

```
$module = new Vtiger_Module();  
$module->name = 'ny_modul';  
$module->save();  
$module->initTables();  
$menu = Vtiger_Menu::getInstance('Tools');  
$menu->addModule($module);
```

*Figur 1. PHP-kod för skapande av en ny modul i vTiger CRM 5.1.*

Övriga åtgärder som krävs för att en ny modul skall fungera är att skapa en ny underkatalog för sin nya modul, i katalogen ”Modules” som finns i vTiger:s rotkatalog. När modulen är skapad kommer användaren åt den nya modulen via vTiger:s programmeny.

## 2.3 HTML

HTML står för Hyper-Text Markup Language och utvecklades av Tim Berners Lee, en fysiker och nätverks specialist som även utvecklade HTTP (Hyper Text Transfer Protocol). HTML är som namnet antyder, ett språk som gör det möjligt att märka upp webbtex, så att det presenteras på ett visst sätt, man pratar också om textformatering.

Språket består av ett stort antal HTML-element, dessa styr formateringen och struktureringen av text för en webbsida. (Pfaffenberger, Karow, White, Schafer 2003: 4-5)

Ett problem med HTML är att olika webbläsare kan tolka samma HTML-kod på lite olika sätt, därför har man redan från märkspråkets begynnelse, strävat efter att standardisera HTML. Detta görs genom W3C, *World Wide Web Consortium*, som är ett industrikonsortium grundat år 1994 och består av ett stort antal företags representanter från hela världen. (Pfaffenberger m.fl. 2004: 8-9)

HTML är plattformsoberoende och ställer inga krav på särskilda verktyg, vilken text-editor som helst duger för att skapa HTML-dokument. Filändelsen för dokumentet skall vara ".htm" eller ".html".

```
<html>
<head>
</head>
<body>
</body>
</html>
```

Figur 2. Standardstruktur för ett HTML-dokument.

## 2.4 CSS

CSS, eller *Cascading Style Sheets* tillämpas för att skilja på innehåll och formatering för dokument på webben. Utan CSS är webbsideutvecklaren tvungen att för varje HTML-element, skilt ange grafiska attribut såsom färger, bakgrund och storlek, vilket leder till att källkoden för en webbsida blir väldigt lång och oläslig.

```
#stil1 {  
    background: #C0C0C0;  
    color: #F78FFF;  
}  
  
#stil2 {  
    background-image:url('bild1.gif');  
    color: #000000;  
}
```

*Figur 3. CSS-stilmall som anger bakgrunder och textfärger.*

Det finns flera möjligheter att tillämpa CSS; internt och externt. Internt innebär att CSS-koden bäddas in i själva HTML-dokumentet, medan extern tillämpning syftar till CSS som en fristående fil som man sedan hänvisar till i HTML-dokumentet. Den kanske främsta fördelen med externa CSS filer är att det effektiviserar underhållet av webbsidor, eftersom ändringar inte behöver göras för varje enskild HTML-fil, utan det räcker med en uppdatering i en CSS-fil, och samtliga HTML-filer påverkas, därav namnet "Cascading" i akronymen CSS. (Pfaffenberger m.fl. 2004: 369-372)

Utöver formatering av textinnehåll för webbsidor används CSS också för att skapa layouter, det innebär i praktiken att positioneringen av HTML-element styrs med hjälp av CSS. Det finns 3 typer av CSS-positionering:

- Statisk
- Relativ
- Absolut

De ovannämnda positioneringsteknikerna syftar till hur HTML-elementen placeras i relation till övriga element.

Precis som för HTML, så är det standardiseringsorganet W3C som underhåller standardiseringen av specificeringarna inom CSS. CSS har utkommit i flera versioner, när detta lärdomsprov skrevs, var CCS3 inte ännu en vedertagen standard, men har varit under utveckling sedan december 2005.

## **2.5 JavaScript**

JavaScript är ett objektorienterat skriptspråk som länkar samman objekt och resurser på klientsidan, i praktiken betyder detta att JavaScript-kod exekveras i webbläsaren hos en klient. JavaScript är designat för att skapa dynamiska webbapplikationer och tillämpas i kombination med HTML.

JavaScript fungerar på de allra flesta plattformar (Windows, Linux, Mac m.m.). Det krävs inga särskilda verktyg för att använda JavaScript, man kan skriva JavaScript i vilken textredigerare som helst och det enda som behövs för att exekvera skriptet är en webbläsare.

JavaScript är ett skriptspråk som inte direkt kan jämföras med programmeringsspråket Java, det är likt i det hänseendet att det är ett objektorienterat språk, men JavaScript riktar sig till en bredare användargrupp, eftersom det är lätt att lära sig och är väldigt avskalat. Den stora skillnaden mellan Java och JavaScript, är att Java-kod kompileras före den kan exekveras, medan JavaScript är ett interpreterat programmeringsspråk som betyder att koden tolkas i realtid när skriptet exekveras. (Easttom 2007: 11)

JavaScript hette LiveScript från början och introducerades i webbläsaren Netscape. 1995 tog Sun Microsystems över utvecklingen av LiveScript och skriptspråket fick då namnet JavaScript.

JavaScript inkluderas i HTML-dokument antingen genom att skriptet placeras i dokumentet, eller så placeras det i en extern fil som man refererar till i HTML-dokumentet. (Easttom 2007: 1-7)

```
<HTML>
<HEAD>
<!-- INBÄDDAT SKRIPT !-->
<script type="text/javascript">
alert("Mellan script-elementen kommer koden");
</script>

<!-- EXTERN FIL !-->
<script type="text/javascript"
src="scriptfil.js"></script>
</HEAD>
<BODY></BODY>
</HTML>
```

*Figur 4. JavaScript inbäddat i ett HTML-dokument, och hänvisning till extern JavaScript-fil.*

## 2.6 AJAX

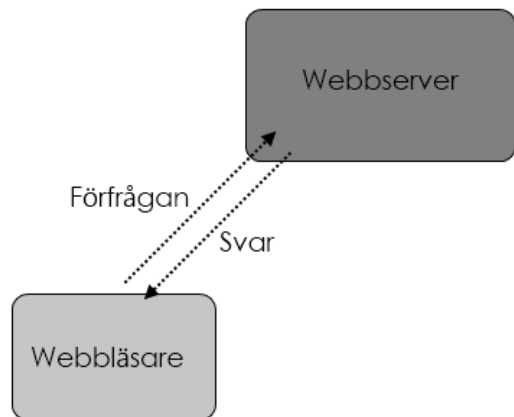
AJAX står för "Asynchronous JavaScript and XML". Ur namnet kan man utläsa att AJAX är en tillämpning av flera tekniker.

AJAX används vanligtvis för att hämta och skicka information i realtid utan att en webbsida som tillämpar tekniken behöver uppdateras helt och hållet. Denna teknik tillför ett beteende hos webbsidor som liknar vanliga skrivbordsapplikationer, med den likheten att bearbetning av data hos webbservern kan ske så att säga i bakgrunden, med den fördelen att det inte stör användbarheten för användaren.

Det traditionella sättet en webbsida visas i en webbläsare följer principen "request/response" (förfrågan/svar), d.v.s. klienten (webbläsaren) skickar en sidförfrågan till en webbserver och den svarar med att leverera webbside-innehållet

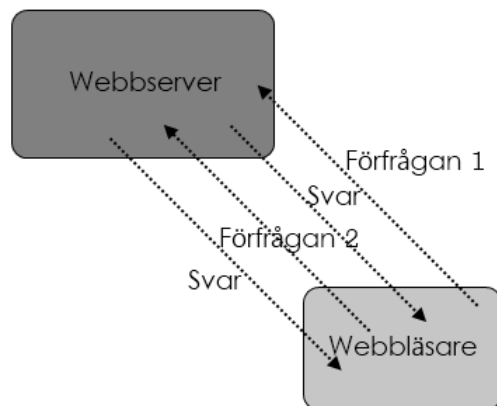


tillbaka till klienten. Datautbytet mellan klient och server sker över HTTP (HyperText Transfer Protocol).



*Figur 5. Princip för ett traditionellt datautbyte mellan klient och server över HTTP-protokollet.*

Hur AJAX rent praktiskt tillämpas är att man utnyttjar det JavaScript baserade XMLHttpRequest objektet för att få klienten att utföra GET/POST anrop mot webbservern på ett asynkront vis utan en total omladdning av webbsidan i fråga.



*Figur 6. Princip för multipla asynkrona förfrågningar.*

Ett praktiskt exempel på fördelar med AJAX är e-post tjänsten Gmail, där man kan ladda upp bilagor för ett e-post meddelande och samtidigt fortsätta skriva sitt meddelande. I samma exempel, i ett scenario där AJAX *inte* tillämpas, hade man varit tvungen att vänta på att bilagan skickats till en webbserver, och att servern svarar med ett klartecken att filuppladdningen genomförts korrekt, förrän man kan fortsätta skrivandet.

Microsoft Internet Explorer skiljer sig från Mozilla Firefox när det gäller att instansiera XMLHttpRequest objektet, därför måste man alltid anpassa sin AJAX-kod så att alla webbläsare stöds. Olikheten beror på att Microsofts webbläsare skapar objektet som en s.k. "ActiveX control", medan alla andra kända webbläsare skapar objektet genom JavaScript. (Babin 2007: 17)

```
if (window.ActiveXObject){  
    xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");  
} else {  
    xmlhttp = new XMLHttpRequest();  
}
```

Figur 7. JavaScript-kod för instansiering av XMLHttpRequest objektet.

De asynkrona förfrågningarna skickas och tas emot med HTTP-metoderna GET eller POST, och svaret som returneras från webbservern tas emot antingen i textformat eller XML-format. (Babin 2007: 16-17)

## 2.7 PHP

Förkortningen kommer ursprungligen från *Personal Home Page tools*, som senare byttes ut till en rekursiv akronym; PHP: *Hypertext Preprocessor*.

Mannen bakom skriptspråket är dansken Rasmus Lerdorf, som år 1994 lade grunden till PHP då han skrev en samling Perl-skript som kunde få en webbserver att generera

dynamiskt innehåll för webbsidor. Ett år senare skrevs skriptsamlingen om i programmeringsspråket C av samma man.

I och med det faktum att PHP är skrivet i språket C, så är syntaxen för PHP väldigt lik, och för dem som har förkunskaper i C-programmering är det mycket enkelt att lära sig det. Från och med PHP version 4 har PHP haft stöd för objektorientering.

PHP hör under kategorin fri programvara och är licensierad under en egen licens som kort och gott heter *PHP License*. PHP är fritt tillgängligt på de vanligaste plattformarna såsom UNIX, Microsoft Windows och MacOS.

I vanliga fall används PHP för att generera dynamiskt innehåll på webbsidor, men PHP används också numera för att skapa skrivbordsapplikationer. PHP har integrerat stöd för ett stort antal databashanterare, men den vanligaste kombinationen är PHP och MySQL.

PHP är ett interpreterat programmeringsspråk, vilket i praktiken betyder att webbservern tolkar PHP skriptet vid exekvering och resultatet av koden skickas tillbaka till webbläsaren som HTML-innehåll.

För att använda PHP, behövs inga andra verktyg utöver en text editor, ytterligare information om programutvecklingsverktyg behandlas i kapitel 4.1.

```
<HTML>
<HEAD>
</HEAD>
<BODY>
<?php
$ta11 = 2;
$ta12 = 5;
$summa = $ta11 + $ta12;
echo "summa: $summa";
?>
</BODY>
</HTML>
```

*Figur 8. PHP-skript bäddas in i ett HTML-dokument. Exemplet visar hur 2 variabler adderas, varefter svaret lagras i en 3:e variabel och skrivs ut på skärmen.*

## 2.8 Databas – MySQL

Databaser används för att lagra data på ett organiserat sätt. Det finns olika typer av databaser, men den vanligaste typen är relationsdatabasen. En av de vanligaste relationsdatabashanterarna idag är MySQL och används av storföretag såsom Google, Cisco, och Nokia. Den främsta skillnaden mellan den och andra stora leverantörers databassystem såsom Microsoft SQL Server är att MySQL finns gratis tillgänglig under licensen ”General Public License”. (MySQL, 2010)

De huvudsakliga grundarna bakom MySQL är finlandssvensken Michael ”Monty” Widenius, och svenskarna David Axmark och Allan Larsson. (MySQL, 2010)

Det finns två olika licensieringar för MySQL, en baserad på öppen källkod (GPL) och en kommersiell variant. Den kommersiella licensen är nödvändig i det fall att det program som använder databasen inte kan publiceras under GPL-licensen.

MySQL använder SQL (Structured Query Language) som är ett standard frågespråk för databaser och används för att hämta ut och modifiera data i en databas. (Vaswani 2005: 11)

```
SELECT fornamn, efternamn, epost FROM personregister  
ORDER BY efternamn ASC
```

*Figur 9. SQL-fråga som hämtar ut alla förnamn, efternamn och e-postadresser ur ett personregister och ordnar dem i stigande ordning enligt efternamn.*

### 3 PLANERING

Den kravspecifikation som först kom från beställaren var mycket kortfattad, ”en veckokalender där användaren kan boka in sålda hus, och mata in kapacitet”.

Med tidigare erfarenhet av liknande projekt i åtanke där kravspecifikationen är kortfattad och vag, stod det klart att arbetssättet för det här projektet måste ske inkrementellt, där kravspecifikationen växer fram och detaljeras med tiden. Lämplig systemutvecklings metodik för att utveckla ett system som detta är ”Extreme programming”, där ledtiderna är korta, och programmet levereras i mindre men funktionerande delar, som med korta mellanrum integreras till allt större helheter. (Wiktorin 2003: 194-196)

Fördelen med programutveckling för webben, är att varje ny ändring man gör, kan snabbt visas upp för beställaren, genom att man laddar upp den senaste versionen på en webbserver, och skickar en länk via e-post till beställaren. Allt som krävs för att beställaren skall kunna utvärdera den senaste versionen eller prototypen, är en webbläsare.

Uppskattad tidsåtgång för detta arbete var 300 timmar, tidsuppskattningen gjordes tillsammans med arbetsledaren och godkändes av beställaren.

#### 3.1 Specificering av krav

Kravspecifikationen detaljerades allteftersom delar av systemet visades upp för beställaren. När en första prototyp var så långt utvecklad att det var möjligt för en användare att mata in och spara reserveringar i en kalender, framkom följande behov:

- Kunderna skall hämtas från CRM:et ur en kundkatalog, och enbart de kunder med vilka det gjorts köpeavtal, skall vara tillgängliga.

- En försäljare skall enbart ha tillgång till sina egna kunder i kundkatalogen, medan företagsledningen skall ha tillgång till alla kunder.
- Kapaciteten skall anges i antalet sammanlagda monteringsdagar per beställning för en viss arbetsgrupp.
- Kapaciteten skall räknas ut automatiskt av systemet, när användaren anger specifikationer för en viss hustyp med tillhörande attribut.
- Företagsledningen vill kunna styra vilka veckor inom vilka det går att göra bokningar, med andra ord en låsfunktion.
- En funktion som ritar upp ett stapeldiagram över ett års totala frikapacitet och upptagna kapacitet.
- En exportfunktion som genererar en årsrapport i Microsoft Excel-format.
- Företagsledningen skall kunna radera reserveringar, samt kunna flytta en reservering från en vecka till en annan.
- Veckornas status skall indikeras med färgerna röd, gul och grön.
- När en användare försöker reservera ett husbygge för en vecka vars reserverade kapacitet kommer att överstiga den lediga kapaciteten, skall det vara möjligt att överskrida gränsen en gång, varefter den låses för ytterligare reserveringar.

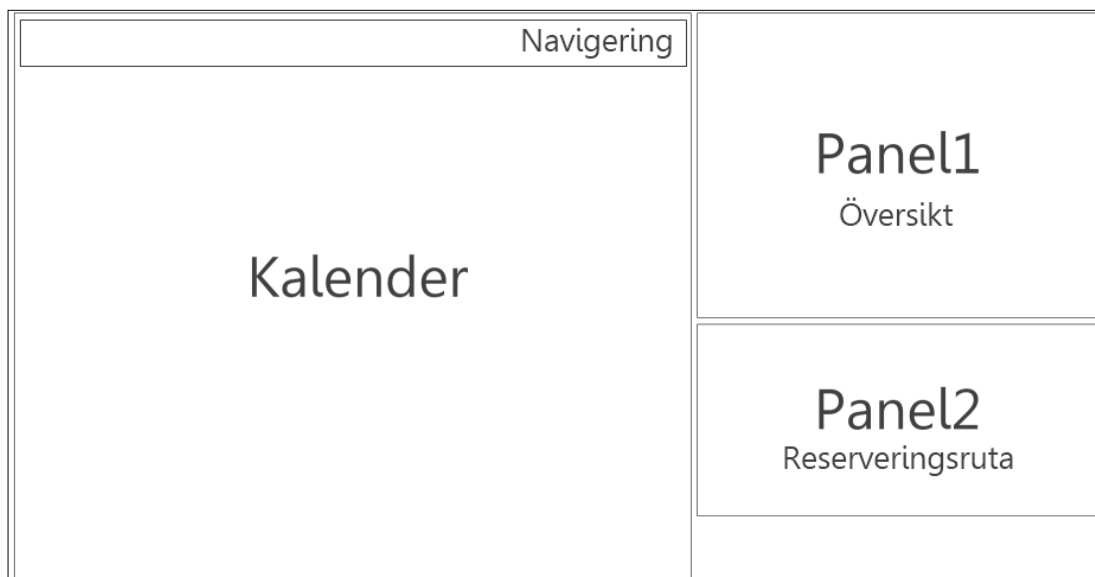
### **3.2 Målgrupp**

Eftersom CRM:et vTiger tidigare hade införts i företagets verksamhet, så fanns en hierarkisk indelning av Jetta-talo:s arbetstagare för vilka det hade tilldelats användarroller som man med fördel kunde dra nytta av i fråga om

användarrättigheter. I en grov målgruppsanalys kunde två typer av användarroller urskiljas; företagsledning och försäljare. Ur ett tekniskt perspektiv kunde man också i och med CRM:et fastslå att alla inom målgruppen för systemet har en webbläsare, med stöd för JavaScript av den orsaken att CRM systemet kräver det.

### 3.3 Användargränssnitts design

Krav eller önskemål kring layouten för systemet fanns inte. En skiss för hur webbtjänstens layout gjordes, men även om fria händer gavs, så låg fokus på enkelhet och minimalism till förmån för användbarhet.



*Figur 10. Skiss för webbtjänstens layout.*

I planeringsskedet för webbtjänsten utseende togs beslutet att positioneringen av de olika panelerna skulle ske med s.k. flytande CSS-positionering, där Panel1 och Panel2 så att säga flyter till vänster om kalenderpanelen. Eftersom denna webbtjänst var tänkt att kunna visas i varierande slag av skärmstorlekar, togs också det beslutet, att alla layoutelement skulle ha en relativ bredd, som anpassas efter användarens



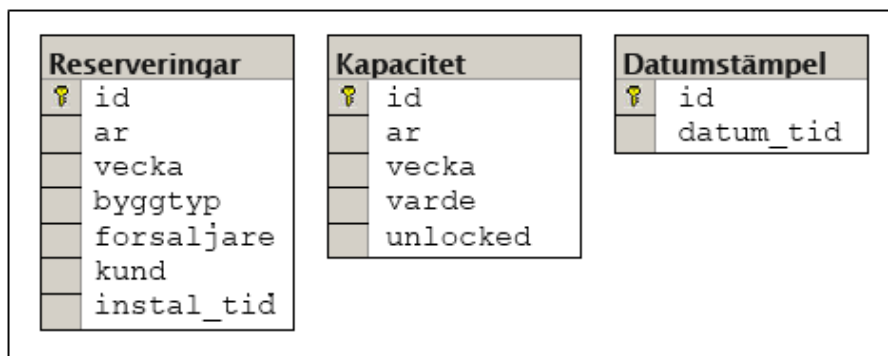
dataskärm. I praktiken betyder det att alla bredlekar anges med ett procenttal, istället för ett bestämt antal bildpunkter.

### 3.4 Databas design

En databas behövdes för att lagra alla reserveringar och för att hålla reda på hur mycket fri kapacitet samt reserverad kapacitet varje vecka innehåller.

Eftersom CRM:et vTiger är konstruerat för att användas tillsammans med databashanteraren MySQL, så föll valet av databas naturligt på MySQL.

Databasen för detta system består av 3 stycken tabeller. Utöver dessa tabeller används också CRM:ets databastabeller, som lagrar kunddata, användarroller o.s.v. men detta behandlas inte i denna avhandling.



*Figur 11. Tre stycken databastabeller. Av datasäkerhetsskäl är namnen för tabellerna och fälten inte de samma som i verkligheten.*

Reserveringstabellen lagrar information om husreserveringar, kapacitetstabellen lagrar de veckor för vilka det angetts frikapacitet för samt ifall en viss vecka är öppen eller låst. Tabellen datumstämpel har till uppgift att hålla reda på när senaste förändring har gjorts i systemet. Samtliga tabeller har en id-kolumn som är tilldelad en primärnyckel.

### 3.5 Skottår och veckoantal

Vart fjärde år är det skottår så en kalender är således inte helt linjär från år till år. Skottår innebär att februari är 1 dag längre vart fjärde år, detta för att kompensera för förskjutning av årstiderna. Beroende på hur december månadens sista dag förefaller, varierar antalet veckor per år mellan 52 och 53. Om vecka 52 slutar före den 28 december, tillkommer en extra vecka. Eftersom datum och tid kan uttryckas på lite olika vis länder emellan så finns det en standard för datumformat och tidsintervall för att eliminera missförstånd, standarden heter ISO-8601. (ISO standarder, 2010)

En undersökning i PHP manualen och en prototyp gjordes i PHP för att bekräfta att skottår tas i beaktande och att veckonumreringen beter sig som förväntat inom programmeringsspråket. Enligt manualen har PHP följt ISO-8601 standarden för veckonumrering sedan version 4.1.0. (PHP manual, 2010) Genom prototypen kunde också detta bekräftas. (prototyp i bilaga 1)

## **4 GENOMFÖRANDE**

### **4.1 Verktyg**

Under denna rubrik redogörs de verktyg som valts inför projektet. För att skapa en webbtjänst som denna användes ingen proprietär mjukvara, det som fordrades var en text editor, ett grafiskt ritprogram och ett filöverföringsprogram.

#### **4.1.1 Notepad++**

Notepad++ är ett enkelt textredigerings verktyg likt Microsoft Notepad (Anteckningar) som är ett standard program i samtliga Microsoft Windows operativsystem. Text editorn är anpassat för en väldigt bred repertoar programmeringsspråk, exempel på funktionalitet är automatisk komplettering av funktioner och syntaxmarkering med olika färger. En viktig aspekt i valet av textredigerare för utveckling av all typ av material för webben är behovet av att kunna välja olika teckenuppsättningar och format för de filer man skapar, detta är inte möjligt i t.ex. Microsoft Notepad.

#### **4.1.2 Paint.NET**

Paint.NET är ett grafikprogram för skapande av all typ av rastergrafik, den mest betydande funktionaliteten för skapande av grafik för webbdesign, är att programmet kan hantera lager och genomskinlighet. Det främsta argumentet för att använda detta program är att det är fritt att använda, och kräver inga licenser, i jämförelse med t.ex. Adobe Photoshop, som skolan visserligen ger tillgång till, men särskilda licenser krävs för att använda den grafik i kommersiella syften som skapas genom det programmet.

### 4.1.3 FileZilla FTP

För att kunna ladda upp HTML-dokument, PHP-skript e.t.c. till beställarens webbserver, så krävs ett klientprogram för överföring av filer över Internet. Ett vanligt sätt att skicka filer över nätet är att använda ett FTP-(File Transfer Protocol) program.

För detta projekt har ett program vid namn *FileZilla FTP* använts. FileZilla är ett verktyg med ett grafiskt användargränssnitt som stödjer aktiva och passiva anslutningar, har automatisk igenkänning av filserver-inställningar, men det främsta argumentet för att välja detta program är att FileZilla är baserat på öppen-källkod och därmed helt gratis. Ibland är det nödvändigt att justera filrättigheterna för vissa kataloger eller enskilda filer, och då är det tacksamt att ha ett grafiskt användargränssnitt för ändamålet. Programmet finns i portabelt utförande, vilket innebär att det kan exekveras från ett USB-minne.

## 4.2 Utvecklingsmiljö

Webbservern som använts i utvecklingen bestod av en "Linux-dator" med operativsystemet Ubuntu 10.04 LTS som hade servermjukvarorna Apache, MySQL och PHP installerade - en programkombination som också brukar kallas LAMP.

För att säkerställa att tjänsten även fungerar på andra plattformar, t.ex. om beställaren byter webbserver-leverantör, så installerades också en s.k. WAMP-server på en arbetsstation med Microsoft Windows 7.

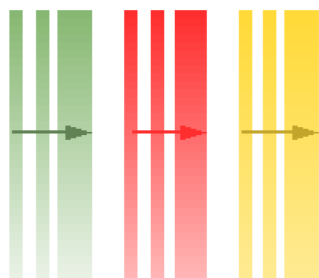
## 4.3 Grafik

Grafiken som använts är egenhändigt skapad i Paint.NET, förutom ett antal programikoner som fritt får användas och modifieras enligt *Creative Commons Attribution 3.0* licensen.

Grafiken skapades med korta laddningstider i åtanke, med andra ord, så liten filstorlek som möjligt. Bakgrundsgrafik för de olika panelerna och tabellerna består av till formatet, smala bildfiler som repeteras horisontellt med hjälp av CSS.

```
.td_cal_green {  
  background-image:url('gfx/cellbg_green_over.gif');  
  background-repeat: repeat-x;  
}
```

*Figur 12. Bakgrundsrepetering med CSS.*



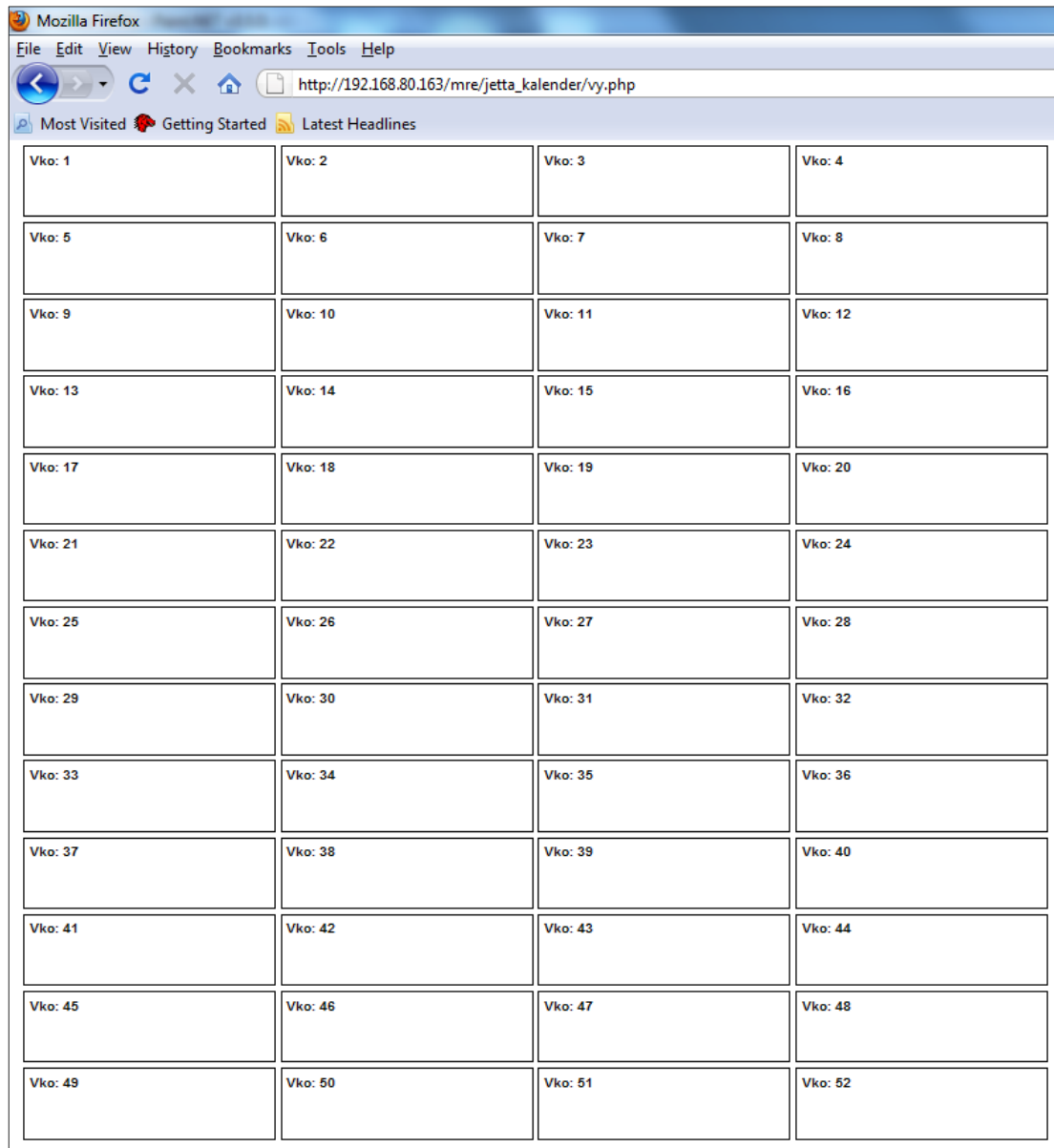
*Figur 13. Princip för repetering av bakgrundsgrafik.*

Varje veckoruta har en viss bakgrundsfärg beroende på vilken status den har. Färgvalen för färgindikationen gjordes i linje med vad beställaren önskade:

- Grön  
Ledig veckoruta, reserveringar tillåtna.
- Gul  
Låst veckoruta, inga reserveringar tillåtna.
- Röd  
Vecka fullbokad, inga fler reserveringar tillåtna.

#### **4.4 Förverkligande av funktioner**

Det praktiska arbetet inleddes med att digitalisera en kalender i veckoformat. Detta gjordes genom en s.k. ”loop” (iteration) som ritar upp 52 rutor i form av en tabell där varje cell motsvarar en vecka. Vid var fjärde veckoruta görs en radbrytning. I detta skede togs det beslutet att ett år visas åt gången, och sedan kan användaren bläddra mellan åren, via framåt- och bakåt-knappar. Anledningen till detta val var att programmet skall kunna visas i ett fönster, i de flesta skärmstorlekar, utan att användaren behöver ”skrolla” (rulla ner) fönstret eftersom det kan störa helhetsöversikten.



*Figur 14. En första prototyp av en kalender.*

Efter att rutorna var uppbyggda fortsatte arbetet med att skapa en första CSS-mall för systemet, där kalendern placerades i centrum av webbläsarfönstret, och sedan lades 2 mindre paneler till, för kommande interaktions möjligheter.

Vuosi: 2010				Seuraava vuosi »	Kapasiteetti - Yhteenveto
Vko: 1 Status:	Vko: 2 Status:	Vko: 3 Status:	Vko: 4 Status:		
Vko: 5 Status:	Vko: 6 Status:	Vko: 7 Status:	Vko: 8 Status:		
Vko: 9 Status:	Vko: 10 Status:	Vko: 11 Status:	Vko: 12 Status:		
Vko: 13 Status:	Vko: 14 Status:	Vko: 15 Status:	Vko: 16 Status:		
Vko: 17 Status:	Vko: 18 Status:	Vko: 19 Status:	Vko: 20 Status:		
Vko: 21 Status:	Vko: 22 Status:	Vko: 23 Status:	Vko: 24 Status:		
Vko: 25 Status:	Vko: 26 Status:	Vko: 27 Status:	Vko: 28 Status:		
Vko: 29 Status:	Vko: 30 Status:	Vko: 31 Status:	Vko: 32 Status:		
Vko: 33 Status:	Vko: 34 Status:	Vko: 35 Status:	Vko: 36 Status:		
Vko: 37 Status:	Vko: 38 Status:	Vko: 39 Status:	Vko: 40 Status:		
					Panel 2

*Figur 15. Kalendern med tillhörande paneler.*

För att förtydliga att rutorna föreställer veckorutor i en kalender, skrevs en funktion som skriver ut veckonummer och ett datumintervall för varje vecka. Just datumintervallet var det kanske mest tidskrävande delmomentet i programmeringen, därför redogörs den mer i detalj i slutet av denna rapport. (kodexempel och figurer i bilaga 2)

För att förtydliga interaktionsmöjligheterna för veckorutorna, lades en s.k. ”mouse-over” effekt till varje ruta, samt att muspekaren indikerar klickbart innehåll. Dessa effekter implementerades genom CSS.



```
.td_cal_green:hover {  
    background-image:url('gfx/cellbg_green_over.gif');  
    background-repeat: repeat-x;  
    cursor: pointer;  
}
```

*Figur 16. Mouse-over och muspekareffekt för tabell celler i CSS.*

#### 4.4.1 Navigeringsfunktionalitet för ändring av årtal

För att skapa navigeringsfunktionalitet för byte av årtal, användes en kombination av PHP och JavaScript. Navigeringen mellan årtal går principiellt till på det viset att användaren klickar på framåt eller bakåt knappar enligt figuren nedan:



*Figur 17. Navigeringsfunktion för byte av årtal.*

PHP skriptet skriver ut HTML-innehåll med framåt/bakåt knapparna. Knapparna reagerar på klickning och gör en ny sidförfrågan till webbservern genom JavaScript, och samtidigt skickas det aktuella året *minus* eller *plus ett*, med i webbadressen. Kalendern laddas om och det valda året som användaren vill visa, fångas upp som en s.k. GET-variabel.

```

<?php
echo "<input type=\"button\" value=\" &#171; \"
onClick=\"document.location.href = '?y=\".($vuos-1) '\"
; return false\"/>
$vuos
<input type=\"button\" value=\" &#187; \"
onClick=\"document.location.href = '?y=\".($vuos+1) '\"
; return false\"/>";
?>

```

*Figur 18. Kombination av JavaScript och PHP, navigerings funktionalitet för byte av årtal. Variabeln "vuos" innehåller det aktuella året i kalendervyn.*

#### 4.5 Reserveringspanel

Nästa steg i utvecklingen var att bygga upp en panel där försäljarna reserverar sålda hus. Informationen som lagras för en reservering är följande parametrar:

- Veckonummer
- Årtal
- Byggnadstyp
- Försäljare
- Kund
- Monterings dagar

Grundtanken var den att så litet information som möjligt skall skrivas in manuellt av användaren, fördelarna med den typen av datainmatning, är att desto större frihet användaren ges, ju större är risken att programfel uppstår i och med felaktiga inmatningar.

Veckonummer och årtal väljs genom att användaren klickar på en vecka i kalendern och informationen skickas till reserveringspanelen.

Byggnadstyperna är begränsade till 4 olika huvudtyper: egnahemshus, parhus, radhus och garage. Byggnadstypen väljs genom en rullgardinsmeny. Försäljarens namn hämtas automatiskt från inloggningssessionen, och kan inte ändras manuellt. Kunden väljs ur en kundkatalog som öppnas genom att användaren klickar på en länk som finns bredvid fältet. Kundfältet är också låst för manuell editering. Antalet monteringsdagar för ett specifikt husbygge, får användaren reda på genom att öppna upp ett kalkyleringsfönster som aktiveras genom en länk bredvid fältet.

The screenshot shows a web-based reservation form. At the top, there are navigation buttons: a left arrow, a text field containing '[0/0]', a right arrow, and a 'Poista' button. To the right, there are dropdown menus for 'Vuosi: 2011' and 'Vko: 18'. The main form area contains four labeled input fields: 'Rakennustyyppi:' with a dropdown menu showing 'OKT'; 'Myyjä:' with a text field containing 'Fredrik Försäljare'; 'Asiakas:' with a text field containing 'Kalle Kund' and a link icon labeled 'Asiakkaat'; and 'Asennus päiviä:' with a text field containing '10.5' and a link icon labeled 'Laskuri'. At the bottom of the form are two buttons: 'Tallenna' and 'Peruuta'.

Figur 19. Skärmbild av reserveringspanelen.

#### 4.5.1 Borttagningsfunktion

Ifall någon användare matar in felaktiga uppgifter behöver det finnas möjlighet att radera en post, detta görs genom en knapptryckning, den post som för tillfället är aktiv i bläddringsläget raderas, men före den tas bort får användaren ännu en chans att ångra sitt val, genom en s.k. ”bekräftelse ruta” där man antingen bekräftar borttagningen eller avbryter processen. Borttagningsfunktionen är endast tillgänglig för användare med administratörsrättigheter enligt önskemål från beställaren.

### 4.5.2 Bläddringsfunktion

När en användare väljer en vecka från kalendern där det redan finns en eller flera reserveringar, kan användaren se detta genom att antalet poster per vecka visas inom parentes i reserveringsrutans övre vänstra hörn. Parentesen omges av bläddringsknappar, med vilka användaren kan stega framåt och bakåt för att se vilka reserveringar som finns.

Tekniken som ligger till grund för bläddringsfunktionen är en kombination av AJAX och PHP. När användaren klickar på en veckoruta, skickas en AJAX-förfrågan tillsammans med parametrar för vilken vecka och vilket år som är valt, till ett PHP-skript som i sin tur gör en databas förfrågan. De reserveringar som finns i databasen skrivs ut som en sträng med separeringstecken mellan varje post. AJAX funktionen fångar upp textsträngen från PHP-skriptet när PHP-skriptet har laddats klart, och genom tillståndsvariabeln "responsetext" skickas textsträngen vidare till en JavaScript funktion som bryter upp strängen enligt separeringstecknet och lagrar alla de reserverade posterna i en global vektor som är åtkomlig för övriga funktioner. När användaren bläddrar bland reserveringarna, hämtas informationen ur vektorn. För att inte förorsaka onödiga AJAX förfrågningar, är systemet uppbyggt på det sättet, att en inledande kontroll efter reserveringar görs när sidan laddas, och endast de veckorutor som har reserveringar kan anropa AJAX-funktionen.

### 4.5.3 Spara och Avbryt

För att spara en reservation klickar användaren på sparknappen. För att avbryta en pågående reservation kan användaren antingen klicka på avbryt eller klicka på en annan veckoruta så töms all inmatad data. Reserveringspanelen omsluts av ett formulär-element som vid ett klick på sparknappen skickas vidare genom metoden POST till ett PHP-skript som ansluter till databasen och lagrar informationen i databastabellen för ändamålet. När lagringen är klar skickas användaren tillbaka till

kalendervyn, och det år som var aktiverat vid sparandet skickas också med i webbadressen så att användaren kan fortsätta på där hon slutade.

#### 4.5.4 Tillgängliga och otillgängliga fält

Beroende på vilka val användaren gör varierar inmatningspanelens tillgänglighet vad gäller knappar och textfält, detta för att undvika programkörningsfel. Ett exempel är sparknappen, som inte skall kunna anropa sparfunktionen utan att någon vecka är vald. Att styra tillgängligheten för knappar och textfält är möjligt genom JavaScript, som kan hänvisa till och manipulera inmatningskontrollerna genom webbsidans DOM (Document Object Model)-objekt och variera attributet "disabled" för dessa. Eftersom det fanns många olika scenarion där tillgängligheten slås på och av, skapades en funktion för ändamålet.

```
function dsbl_form() {  
  var elem = document.getElementById('frm1').elements;  
    for(var i = 0; i < elem.length-1; i++) {  
      elem[i].disabled=true;  
    }  
}
```

*Figur 20. JavaScript som varierar tillgängligheten för inmatningspanelens inmatningskontroller.*

Det som händer i funktionen som visas i kodexemplet ovan är att hänvisningar till inmatningskontrollerna inom det formulär ("frm1") som omsluter reserveringspanelen, lagras som objektreferenser i vektorn "elem". Efter detta sätts attributet "disabled" för alla inmatningskontroller genom en vektor iteration.

## 4.6 Översiktspanel

För att snabbt kunna ge användaren en överblick av reserveringarna, gjordes en skild panel för detta, innehållande en tabell där upptagen kapacitet och fri kapacitet ställs mot varandra, med mellansummor och totalsummor.

Kapaciteetti - Yhteenveto		
Viikot	Varattu	Vapaa
1, 2, 3, 4	100	80
5, 6, 7, 8	0	160
9, 10, 11, 12	0	160
13, 14, 15, 16	0	190
17, 18, 19, 20	30	200
21, 22, 23, 24	0	200
25, 26, 27, 28	0	200
29, 30, 31, 32	0	200
33, 34, 35, 36	125	200
37, 38, 39, 40	0	200
41, 42, 43, 44	0	200
45, 46, 47, 48	0	160
49, 50, 51, 52	0	40

[Näytä pylväsdiagrammi](#)
[Export exceliin](#)

---

**Tot. varattu kapasiteetti:** 255.0  
**Tot. vapaa kapasiteetti:** 2190.0

---

**Viimeisin päivitys tehtaalta: [ 2010-09-24 18:51:47 ]**

Figur 21. Skärmbild av översiktspanelen i sin helhet.

### 4.6.1 Datumstämpel för senaste uppdatering

När företagsledningen justerar frikapacitet eller låser/öppnar någon veckoruta, skrivs en datumstämpel till databasen, för att de övriga användarna snabbt skall kunna se från gång till gång om något har hänt sedan förra gången de använde webbtjänsten.

#### 4.6.2 Excel rapportfunktion

För att företagsledningen skall kunna dela med sig av kapacitets- och reserveringsinformationen till övriga personer som inte har tillgång till detta system, så behövs en funktion som kan skriva ut rapporter i ett format som är gångbart inom branschen. Eftersom Microsoft Excel är ett kalkylprogram som är mycket utbrett inom företagsvärlden föll valet av format naturligt på just detta.

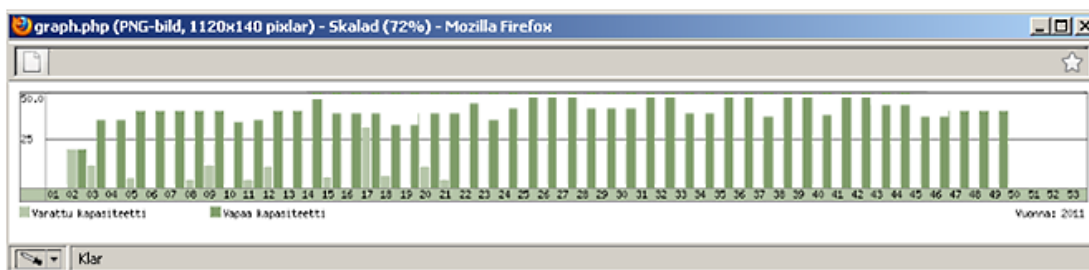
Informationen som skrivs ut i rapporten är samma information som syns i översiktspanelen och därför kunde samma funktioner återanvändas för detta ändamål. För att skapa ett Excel dokument genom programmeringsspråket PHP så behövs inga särskilda tilläggs moduler, det går till så att det s.k. sidhuvudet för en webbsida (HTTP-header) modifieras, så att webbläsaren kan identifiera att informationen som skickats från webbservern är en nedladdningsbar bilaga och ifall användaren har Microsoft Excel installerad på sin dator, så känner operativsystemet av att innehållet är en Excel fil.

#### 4.6.3 Stapeldiagram

Beställaren efterfrågade ett stapeldiagram som visar reserverad kapacitet och frikapacitet för ett helt år i samma diagram. För att möta dessa krav behövdes mjukvara som dynamiskt kan manipulera grafik. Det finns idag många varianter av programbibliotek för generering och manipulering av grafik, som stöds av PHP.

För detta arbete valdes ett programbibliotek som heter *GD Graphics Library*. När man skapar grafik programmatiskt anger man koordinater i form av bildpunkter i bredd och höjd. Detta bibliotek har färdiga metoder för att rita upp streck och rektanglar, vilket i mitt fall var det enda som behövdes för att rita ett stapeldiagram; X- och Y- axlar i form av streck och rektanglar användes som staplar. För att skilja på staplarna användes olika färgnyanser. Diagramfunktionen skriver ut diagrammet i form av en bild i PNG (Portabel Network Graphics)-format, vilket går till på liknande

vis som för Excel-rapport funktionen, - sidinformationen i HTTP-header anger att data som skickas från servern skall tolkas som grafik av webbläsaren.



*Figur 22. Skärmbild av stapeldiagramms funktionen. Reserverad kapacitet och frikapacitet i samma diagram.*





















#### 4.6.4 Kapacitetsgräns

När en försäljare försöker reservera en bokning som överskrider kapacitetsgränsen kommer ett meddelande upp på skärmen, och programmet tillåter användaren spara högst *en* bokning som överskrider gränsen.

#### 4.6.5 Låsfunktion

Beställaren ville att företagsledningen skall kunna bestämma ifall det är möjligt för försäljarna att boka in försäljningar i kalendern, för att styra detta gjordes en låsfunktion där ett öppet och ett stängt hänglås visas bredvid varje veckas kapacitet, som företagsledningen sedan kan öppna och låsa vid behov. För att indikera en låst vecka i kalendern används en gråtonad färg eftersom det kan kännas igen från många andra datorprogram. När någon försöker klicka på en låst vecka får användaren ett meddelande om att veckan för tillfället är låst.



Viikko	Kapasiteetti	Status
1	<input type="text" value="0.0"/>	   
2	<input type="text" value="20.0"/>	   
3	<input type="text" value="40.0"/>	   
4	<input type="text" value="40.0"/>	   
5	<input type="text" value="40.0"/>	   

Figur 23. Skärmbild av låsfunktionen.

#### 4.6.6 Kundkatalog

Vid en reservering av ett nytt husbygge måste en kund knytas till reserveringen. Det går praktiskt till så att kundkatalogen öppnas i ett nytt fönster där en lista på kunder visas, användaren. Dessa hämtas från CRM:ets kunddatabas. Varje kund i kundhanteringssystemet har en ”ägare”, som antingen är en CRM användare eller en grupp av användare. I kundkatalogen används detta som en ett sök kriterium, och bara de kunder som den inloggade användaren har ägarskap över, filtreras ut och visas i listan.

Eftersom det finns kunder i CRM:et med vilka man inte har gjort ett köpbeslut samt har avtal med, så ingår detta faktum också som sök kriterium d.v.s. alla som har skrivit under avtal hör till urvalslistan . Nedan visas ett kodexempel för hur sökningen i databasen går till, uttryckt i frågespråket SQL.

De användare som har administratörsrättigheter har möjlighet att simulera inloggning av en viss försäljare och får därmed tillgång till alla kontakter inom CRM:et. Det som presenteras i kundkatalogen är förnamn, efternamn, e-post och telefonnummer.

```

SELECT vtiger_contactdetails.accountid,
vtiger_contactdetails.firstname,
vtiger_contactdetails.lastname,
vtiger_contactdetails.email,
vtiger_contactdetails.phone, vtiger_contactscf.cf_565
FROM vtiger_contactdetails INNER JOIN vtiger_crmentity
ON
vtiger_contactdetails.contactid=vtiger_crmentity.crmid
INNER JOIN vtiger_contactscf ON
vtiger_contactdetails.contactid=vtiger_contactscf.con
tactid WHERE vtiger_crmentity.deleted = '0' AND
vtiger_contactscf.cf_565 = 'Sopimus tehty' AND
vtiger_crmentity.smownerid = '$userid' ORDER BY
vtiger_contactdetails.lastname;

```

*Figur 24: SQL-satsen som används för att hämta kunder ur CRM:ets databastabeller.*

*Notera att kontakter är det samma som kunder.*

#### 4.6.7 Arbetstids kalkylator

För att förenkla uträkningen av arbetstid som skall reserveras, ville beställaren ha en arbetstidskalkylator, som automatiskt räknar ut arbetstiden, när användaren anger ett antal parametrar som påverkar arbetstidens längd. Parametrarna som styr arbetstiden är följande:

- Byggnadstyp
- Kvadratmetrar
- Antal våningar
- Antal lägenheter
- L-husformering

- Enkelt burspråk
- Mutterformade burspråk
- Byggnadsstomme
- Takläkt
- Vattentak
- Vattentakstyp
- Antal girar
- Årstid (vinter/sommar)

**Asennus-aika laskuri**

OKT Kerrosneliöt: 170

Kerrokset: ☒ 1 ☐ 1.5 ☐ 2

☒ L-malli

☒ Erkkerit Kpt: 2

☐ Mutterierkkerit Kpt:

**Asennuslaajuus**

runko

ruode

vesikatto

deselect/ select multiple using Ctrl-button

**Vesikatto**

pell

tili

Järit Kpt: 2

☐ Talviaika 1.11–31.3

**Tallenna & sulje >**

**Työaikalaskuri**

RUNKO: 28

ERKKERI: 6

M.ERKKERI: 0

VESIKATTO: 0

RUODE: 9

JÄRIT: 8

KATTOTURVA T.: 0

OTSA- & ALUSLAUTA-AS: 0

PANEELIELEMENTIT: 0

SUBTOTAL: 51 tuntia

TOTAL: 6.4 päivää

Klar

Figur 25. En skärmbild av arbetstids kalkylatorn.

Uträkningarna för hur parametrarna påverkar tiden specificerades av beställaren, och bestod av olika index, som baserar sig på statistik från tidigare år. Varje gång en förändring sker i kalkyleringsfönstret, måste uträkningsfunktionen exekveras på nytt och för denna uppgift fanns det två olika sätt att göra detta på, antingen händelsestyrd eller tidsbaserad uppdatering. Av den orsaken gjordes två prototyper för att kunna avgöra vad som fungerar bättre. Den första prototypen var baserad på händelsestyrd uppdatering, där varje textfält eller flervalsoalternativ aktiverar uträkningsfunktionen, men efter att tester gjordes i olika webbläsare framkom det att det fanns kompatibilitetsproblem mellan webbläsaren Microsoft Internet Explorer och övriga webbläsare. Problematiken ligger i att de olika webbläsarna hanterar händelser för inmatningskontroller på olika sätt, och vissa uppdateringar skedde inte i tillräckligt

snabb takt och användbarheten blev lidande eftersom programmet inte upplevdes tillräckligt responsivt. Den andra prototypen med tidsbaserad uppdatering blev betydligt mer användarvänlig, och utifrån ett antal tester valdes ett uppdateringsintervall på 500 millisekunder. Ett för kort intervall utnyttjar onödigt mycket processorkraft hos klienten, medan ett för långt intervall försämrar användbarheten.

```
function tick() {
    var t=setTimeout("calculate()",500);
}
```

*Figur 26. Funktionen "tick()" använder JavaScript metoden "setTimeout" för att med 500 millisekunders intervall köra funktionen "calculate()".*

När uträkningen är klar, bekräftar användaren detta genom att klicka på en sparknapp och det totala antalet arbetsdagarna skickas till reserveringspanelen. Tekniskt görs detta genom en JavaScript funktion som refererar till huvudfönstrets textfält för arbetstid, ger det ett nytt värde och avslutningsvis stängs kalkylator fönstret.

```
function updateParent() {
    var elem =
opener.document.getElementById('frm1').elements;
    elem[9].value = TOTAL;
    elem[9].focus();
    self.close();
return false;
}
```

*Figur 27. JavaScript funktion som från arbetstidskalkylator fönstret skickar data till ett textfält i huvudfönstret för att avslutningsvis stänga kalkylatorn.*

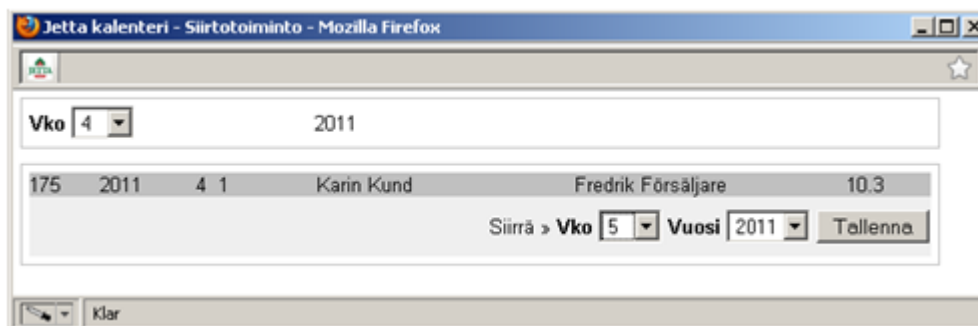
#### 4.6.8 Flyttningsfunktion

Företagsledningens kan av vissa orsaker ha behov av att flytta en reservering från en vecka till en annan, och då behövdes en funktion för detta ändamål. Flyttningsfunktionen öppnas i ett eget fönster, och förflyttningen börjar med att användaren väljer en vecka från en rullgardinsmeny varefter reserveringar för den valda veckan presenteras.



Figur 28. Skärmbild av flyttningsfunktionen, vy 1.

Användaren väljer en reservering genom att klicka på den. De övriga reserveringarna som inte blev markerade för flyttning döljs, och rullgardinsmenyer för vecka och årtal samt en sparknapp blir synliga.



Figur 29. Skärmbild av flyttningsfunktionen vy 2.

När användaren klickar på sparknappen så verkställs förflyttningen i databasen och huvudfönstret uppdateras så att användaren direkt kan se att förflyttningen har utförts.

## **4.7 Säkerhet**

### **4.7.1 Autentisering**

Inloggningen från CRM systemet vTiger genererar en session som hålls vid liv tills användaren loggar ut. En session är en typ av vektor som hålls i arbetsminnet på webbservern. Vissa av sessionsvariablerna som blir initierade genom autentiseringsfunktionen i CRM:et kunde också användas för kalendermodulen, för att säkerställa att ingen som inte har loggat in via vTiger kan få åtkomst till kalendern. Ett exempel på försök till intrång kunde vara att en utomstående person som har erfarenhet om vTiger:s katalogstruktur, skulle försöka komma åt de olika modulerna genom att mata in en absolut sökväg till kalender modulen direkt i adressfältet i sin webbläsare. För att motarbeta detta görs det en kontroll av sessionen i varje enskild PHP-fil, och är inte sessionen rätt initierad så avslutas programmet.

## 5 RESULTAT

Resultatet av detta arbete är en webbapplikation som tillfredställer de funktionella krav som i kravspecifikationen efterfrågats. Programmet har tagits i bruk av beställaren och inga fel har hittills uppdagats. Testningar har gjorts regelbundet i webbläsarna Mozilla Firefox och Internet Explorer efter varje delmoment som implementerats och ingen av webbläsarnas inbyggda felkonsoler har givit fel eller varningar vad gäller JavaScript och syntaxfel.

Detta arbete har givit en praktisk erfarenhet av systemutveckling och planering av ett relativt omfattande projekt innehållande många delmoment och tekniker samspelar och som tillsammans bildar en komplex helhet. (Skärmbild av användargränssnittet från det färdiga systemet i bilaga 3)



## 6 DISKUSSION

Tidsåtgång för det praktiska arbetet uppgick till 330 timmar vilket översteg den uppskattade arbetstiden med 30 timmar, och är enligt mig en relativt marginell skillnad i relation till hur mycket kravspecifikationen växte under projektets gång.

Detta arbete har varit ett motiverande arbete på många sätt, dels är det ett ”riktigt projekt” med en kund som är allmänt känd och så är detta arbete värdefullt för skolan också, eftersom alla genomförda projekt kan användas som referens, som i sin tur leder till flera samarbeten mellan skola och företag.

När jag tänker tillbaka på projektstarten och vad det är idag, kan det ha varit fördelaktigt att inte alla systemkrav inkom samtidigt, eftersom projektet då hade verkat mycket omfattande och komplext, och hade kanske inte blivit av.

Att arbeta enligt agila systemutvecklingsmetoder såsom *Extreme programming* är enligt mig väldigt lämpat för IT-projekt inom webbutveckling. Vad som krävs av programmeraren för att tillämpa utvecklingsmetodiken är framförallt envishet. Den systemutvecklingsmetodik som tillämpats hör under kategorin testdriven utveckling, som innebär att koden ständigt testas efter varje liten förändring. Då är det mycket tacksamt att arbeta med PHP och JavaScript som är skriptspråk, för då krävs ingen kompilering av koden vid varje testning i jämförelse med t.ex. C-programmering. Detta spelar tidsmässigt en stor roll vid utvecklingen när antalet testningar kan uppgå till 100-tals gånger dagligen.

I och med detta arbete har jag fått större insikt i hur värdefull AJAX-tekniken är för datautbyte mellan olika programmeringsspråk, i synnerhet mellan PHP och JavaScript.

Som förslag till vidare forskning vore att undersöka ifall det skulle gå att optimera funktionen som genererar datumintervallen (funktion i bilaga 2) för respektive vecka,

eftersom den är relativt processorkrävande. Dels kunde den data som funktionen genererar, återanvändas på ett mer optimalt vis eftersom funktionen i nuläget exekveras *varje gång* kalendern laddas om vid t.ex. sparning och dels kunde man också undersöka alternativa metoder för att genererar datumintervallen och jämföra de metoder med min lösning och kanske komma till en mer rationell lösning. Som ett resultat av en sådan underökning kunde man eventuellt optimera sidladdningstid i en omfattning av millisekunder.

## **6.1 Iakttagelser under arbetsprocessen**

### **6.1.1 Problem med teckenuppsättningar**

Ett problem som uppstod under arbetets gång och som är viktigt att ta fasta på när man utvecklar för webben, är teckenuppsättningar. Detta är ett mycket vanligt problem framförallt i de Nordiska länderna där man har behov av att använda tecknen ”åäö”. Det enda rätta är att bestämma sig för en teckenuppsättning, och vara konsekvent i alla hänseenden, från databaskopplingen, till vilket format man sparar de enskilda HTML- och skript-filerna. Teckenuppsättningen för alla dessa inställningar bör vara likvärdiga:

- Servers standard teckenuppsättning (default charset)
- Databaskopplingen
- Databastabellen
- Webbssidans teckenformat (i vilket format filen sparas)
- Webbssidans dokumenttyp (doctype)
- Webbssidans sidhuvud (HTTP header)

### **6.1.2 Variation i datumhantering mellan olika plattformar**

Hur datum genereras med PHP skiljer sig mellan Windows-miljö och Linux-miljö fick jag erfara när jag i skolan arbetar i Linux-miljö och fortsatte att arbeta med

projektet hemma i Windows-miljö. Veckonumren räknas med början från 00 till 53 i Windowsmiljö medan det i Linux-miljö räknas från 01 till 53. En annan detalj är att man i Linux-miljön använder formateringsstecknet "%V" för att representera veckonummer, medan det i Windows-miljö finns en motsvarande variant: "%W".

## 7 KÄLLFÖRTECKNING

### Tryckta arbeten

Babin, Lee 2007. Beginning Ajax With PHP - From Novice To Professional. USA. Apress.

Easttom, Chuck 2007. Advanced JavaScript, Third Edition. 3:e upplagan. Texas. Wordware Publishing.

Pfaffenberger, Bryan , Karow, Bill , White, Chuck , Schafer, Steven M. 2003. HTML,XHTML and CSS Bible 3rd Edition. Wiley.

Vaswani, Vikram 2005. How to Do Everything with PHP and MySQL. USA. McGraw-Hill Osborne Media.

Wiktorin, Lars 2003. Systemutveckling på 2000-talet. Studentlitteratur AB.

### Elektroniska publikationer

Gartner, Juli 2009. Statistik över CRM [online]. [hänvisning 26.10.2010]. Tillgänglig i form av www-dokument: <<http://www.gartner.com/it/page.jsp?id=1074615>>

ISO standarder, 2010. Datum och tidsformat [online]. [hänvisning 26.10.2010]. Tillgänglig i form av www-dokument: <[http://www.iso.org/iso/date\\_and\\_time\\_format](http://www.iso.org/iso/date_and_time_format)>

Jetta-Talo webbsida, 2010. Företagsbeskrivning från officiell webbsida. [online]. [hänvisning 26.10.2010]. Tillgänglig i form av www-dokument: <<http://www.jetta-talo.fi/jetta-talooy.html>>

MySQL, 2010. Beskrivning av MySQL, officiell webbsida. [online]. [hänvisning 26.10.2010]. Tillgänglig i form av www-dokument: <<http://www.mysql.com/about/>>

PHP manual, 2010. Datumfunktion [online]. [hänvisning 26.10.2010]. Tillgänglig i form av www-dokument: <<http://php.net/manual/en/function.date.php> >

vTiger webbsida, 2010. Om vTiger CRM [online]. [hänvisning 26.10.2010]. Tillgänglig i form av www-dokument: <[http://vtiger.com/index.php?option=com\\_content&task=view&id=25&Itemid=53](http://vtiger.com/index.php?option=com_content&task=view&id=25&Itemid=53)>

vTLib, 2010. Manual för skräddarsydda moduler [online]. [hänvisning 26.10.2010]. Tillgänglig i form av PDF-dokument: <[http://forge.vtiger.com/frs/download.php/948/VtigerCRM\\_5.1.0\\_Vtlib.pdf](http://forge.vtiger.com/frs/download.php/948/VtigerCRM_5.1.0_Vtlib.pdf)>

## BILAGOR

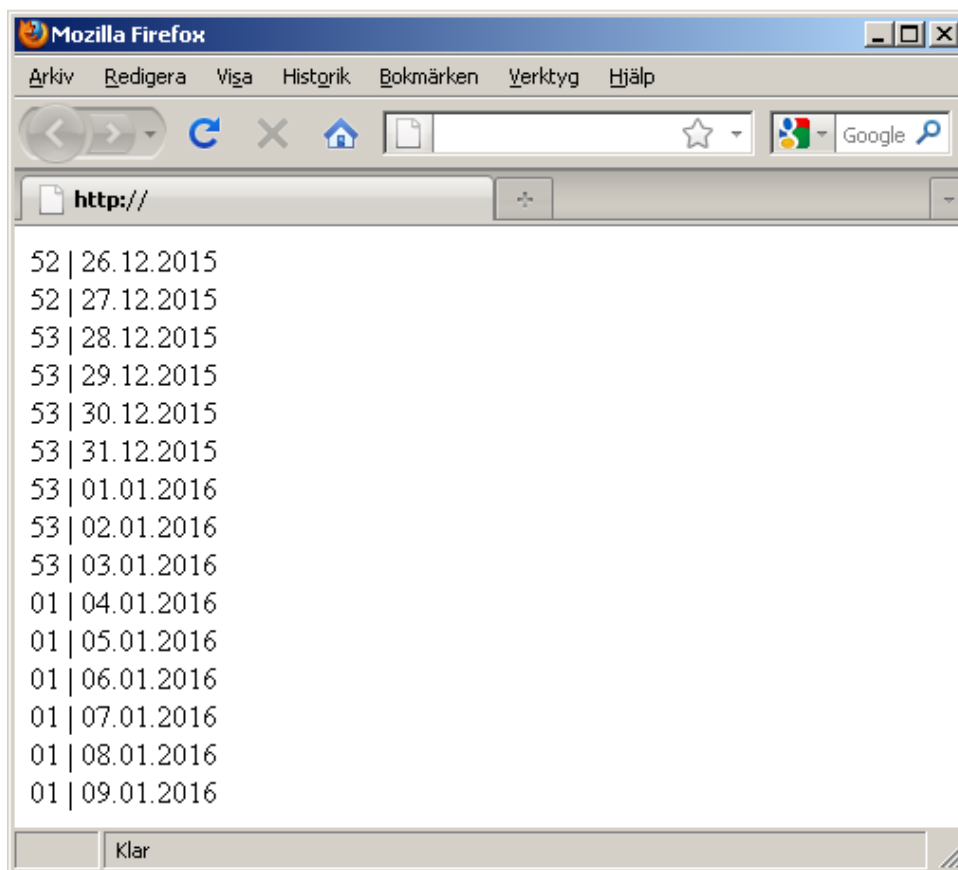
### Bilaga 1. Undersökning av datumhantering i PHP

För att bevisa att programmeringsspråket följer ISO-8601 standarden gjordes ett test i form av ett PHP-skript som genererar ett antal datum tillsammans med veckonummer för dessa.

```
for($i=1;$i<=15;$i++) {  
    $w_nr = strftime("%V", strtotime("12/25/2015 + $i  
day")).  
    " | ".strftime("%d.%m.%Y", strtotime("12/25/2015  
+ $i day"));  
    echo "$w_nr <br/>";  
}
```

*Figur 31. PHP-kod som genererar datum och veckonummer genom en FOR-loop (iteration).*

PHP-funktionen "strtotime()" skapar datum eller tid utifrån en sträng och för att bestämma vilket tidsformat som skall tillämpas, används funktionen "strftime()" som formaterar tiden. Som årtal för testet valdes år 2015, som har 53 veckor.



Figur 32. Testet visar att år 2015 har 53 veckor.

## Bilaga 2. Funktion för datumintervall

```

function vkonr_arr_populate($query_y) {
$vkonnr = array();
global $vkonnr;
///STEG1 Antal veckor för året
if(strftime("%V",
strtotime("01/01/" . (intval($query_y)+1) . ""))=="53") {
$weeksinyear = 53; } else { $weeksinyear = 52; }
///STEG2 Populera datum & vecko vektorerna
$next_y=intval($query_y)+1;
for($i=0;strftime("%d.%m.%Y", strtotime("01/01/$query_y +
$i day"))!="01.01.$next_y";$i++) {
    $date[$i] = strftime("%d.%m", strtotime("01/01/$query_y
+ $i day"));
    $week[$i] = strftime("%V", strtotime("01/01/$query_y +
$i day"));
}
///STEG 3 Klipp början & slutet av året
for($i=0;$i<=10;$i++) {
    if($week[$i]=="$weeksinyear") {
        unset($week[$i]);
        unset($date[$i]);
    }
}
for($i=365;$i>=355;$i--) {
    if($week[$i]=="01") {
        unset($week[$i]);
        unset($date[$i]);
    }
}
///STEG 4 Sammanfoga datumintervall från vecko-vektorn.
for($l=1;$l<=$weeksinyear;$l++) {
    $str="";
    for($i=0;$i<=count($week)+1;$i++) {
        if($week[$i]=="$l" ||
$week[$i]=="0$l") {
            $str = $str.'|'.$date[$i];
        }
    }
    $vkonnr[$l] = substr($str, 1, 5).' -
'.substr($str, -5);
}
if($weeksinyear!=53) { unset($vkonnr['53']); }
return $vkonnr;
}

```



*Figur 33. PHP funktion för att populera en vektor med datumintervall för varje vecka i ett år.*

För att få en inblick i vad denna funktion gör, så är det enklare ifall man delar in funktionen i flera steg. Funktionen tar emot ett årtal som parameter ("\$\_query\_y") och

- **Steg 1**

Inledningsvis kontrolleras ifall det finns 52 eller 53 veckor för angivet år genom att hämta veckonummern för den första dagen, följande år.

- **Steg 2**

I detta steg itereras hela året igenom för att fylla vektorerna "\$date" och "\$week" med datum och veckonummer.

- **Steg 3**

Ifall ett nytt år inte börjar på en måndag eller slutar på en söndag så klipps de dagar bort som antingen flyttats över från föregående år eller som går in i nästa år. Jag valde att göra det på detta vis p.g.a. datumintervallet inte visar årtal och det kan då vara förvirrande att samma datum kan visas på två ställen i kalendern i ett och samma år.

- **Steg 4**

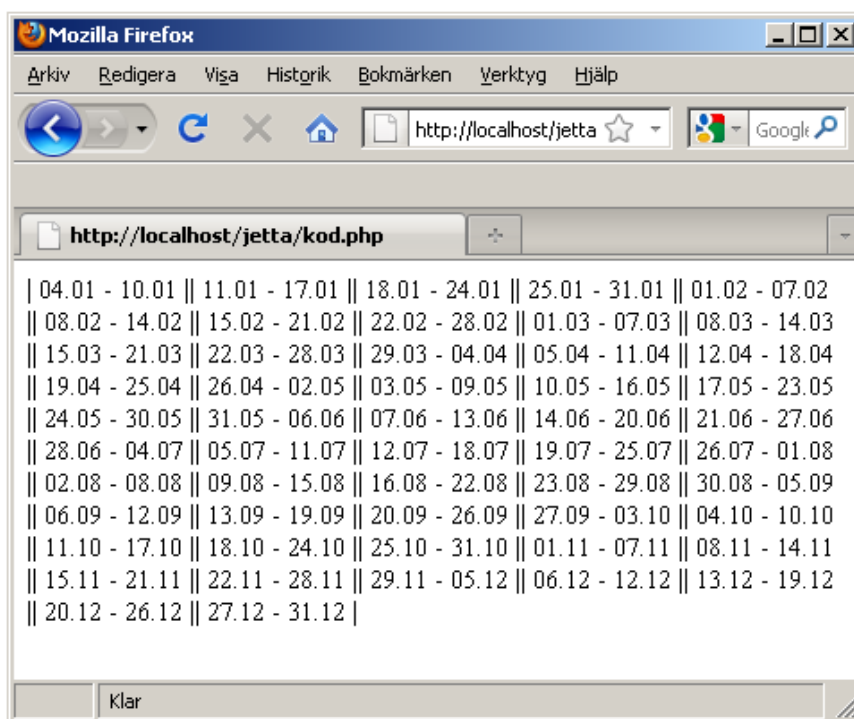
I det fjärde och sista steget sammanfogas veckonumren och datumintervallen på så vis att varje vecka motsvarar ett element i vektorn "\$vkonr" och varje element tilldelas datumintervallet som en sträng i formatet: "dd.mm – dd.mm".

För att bevisa att denna funktion fungerar som förväntat, vill jag illustrera detta genom ett kodexempel samt en skärmbild av resultatet.

```
$vektor = vkonr_arr_populate("2010");  
  
foreach($vektor as $value ) {  
    echo "| $value |";  
}
```

*Figur 34: Tillämpning av funktionen i testsyfte.*

I exemplet skickas parametern "2010" med, när funktionen exekveras och resultatet av programkörningen lagras i en vektor för att slutligen skrivas ut på skärmen genom att vektorn itereras igenom med en foreach-iteration.



*Figur 35. Resultatet av funktionen visad i en webbläsare.*

